

REMARKS

In view of the above amendments and the following remarks, reconsideration and further examination are respectfully requested.

Independent claim 1 has been amended to (i) overcome the 35 U.S.C. § 112 rejections, as discussed below, by clarifying the features of the invention described therein and (ii) further distinguish the claimed invention from the references relied upon in the rejections discussed below.

Claims 1-11 were rejected under 35 U.S.C. § 112, first paragraph for failing to comply with the enablement requirement. Specifically, the rejection states that the term “empty data,” as recited in claim 1, is not supported by the disclosure of the specification. This rejection is clearly inapplicable to amended independent claim 1 and the claims that depend therefrom for the following reasons.

Initially, please note that the specification describes the empty data storage section 23 as being a null device (see paragraph [0028]). Further, it is respectfully submitted that a person of ordinary skill in the art would understand that upon reading data from other devices, a null device does not return any data that is of use (valid) to other devices, but rather, returns an end-of-file (EOF).

For example, in Unix-type operating systems a “null device is a special file that discards all data written to it (but reports that the write operation succeeded), and provides no data to any process that reads from it (it returns EOF)” (see attached Wikipedia article, June, 2008).

Further support for this description of a null device is found in a Linux Programmer’s Manual, which states that “Data written on a null or zero special file is discarded. Reads from

the null special file always return end of file, whereas reads from zero always return \0 characters” (see attached Linux Programmer’s Manual, June, 2008).

Additional support for the above-mentioned description of a null device is found in a Unix programming book, which states “when the data is read from dev/null [the null device], the null device always returns 0 bytes. Therefore, the program which reads data from the null device receives end of file at once” (see attached Unix Programming Environment). _

In view of the above, it is clear that, in response to a read, a null device returns 0 bytes of data that are valid to other devices and also indicates an end of file. Furthermore, in order to clarify this feature of the claimed invention, claim 1 has been amended to recite that the empty data storage section [null device as described in the specification] returns data indicating an end of a file as empty data in response to a data read. This amendment is consistent with the above-cited (and attached) documents as well as the specification.

Therefore, it is clear that, in view of the above-identified descriptions of a null device (empty data storage section), a person of ordinary skill in the art would understand from the specification that (i) the empty data storage section acts as a null device, (ii) the claimed “empty data” is actually data indicating an end-of-file and transmitted from the empty data section (null device), and (iii) the data indicating an end of a file as empty data, as recited in amended claim 1, is data that does not have a header or payload since it is data that is not of use (valid) to the other devices.

Thus, it is respectfully submitted that it would be obvious to a person skilled in the art of the claimed invention that the description in the specification stating “if a data read from the empty data storage section 23 is attempted, only empty data (i.e., 0 bytes) can be read from the

empty data storage section 23. A specific example of the empty data storage section 23 is a null device which is supported by some OS's" merely indicates a conventional technique of a null device returning empty data that is of no use to other devices and that indicates an end of file, which needs no further description. Further, it is respectfully submitted that a person of ordinary skill in the art of the claimed invention would be enabled to make and/or use the claimed invention. Thus, for the reasons discussed above, this rejection is not applicable to claims 1-11.

Claims 1-3 were rejected under 35 U.S.C. § 112, second paragraph as being indefinite. Specifically, the rejection states that it is unclear as to whether the Applicants intend to claim a storage section for empty data or a data storage section that is in fact empty. In view of the above-mentioned amendment of claim 1, it is respectfully submitted that it is clear that the empty data storage section acts as a null device which, in response to a data read, returns data indicating an end of a file as empty data. Thus, in view of the above, withdrawal of this rejection is respectfully requested.

Claims 1-11 were rejected under 35 U.S.C. § 103(a) as being unpatentable over the combination of Olnowich (U.S. 6,092,155) and Kato et al. (U.S. 5,832,215). The above-mentioned 35 U.S.C. § 103(a) rejection is clearly inapplicable to amended independent claim 1 and claims 2-11 that depend therefrom for the following reasons.

Amended independent claim 1 recites a stream data processing apparatus for performing multiple processing steps during a processing of stream data. The stream data processing apparatus of claim 1 includes, in part, an empty data storage section for erasing any data written thereto in response to a data write, and for returning data indicating an end of a file as empty data in response to a data read. Olnowich and Kato, or any combination thereof, fail to disclose or

suggest the above-discussed features of claim 1.

Rather, Olnowich teaches a system which allows the transmission and receipt of null data (see col. 14, lines 49-67), wherein, after data word D_n is transmitted to complete the sending of a valid data word, null (00) words are sent and a valid signal 120 stays active waiting to see if a message is accepted or rejected. In other words, Olnowich teaches that the null words are sent as a payload subsequent to the transmission of a header until it is determined whether the header and payload are accepted/rejected (see Fig. 6).

Thus, in view of the above, it is clear that Olnowich teaches that null data including a header and null (00) words sent as a payload until the data is accepted/rejected, but fails to disclose or suggest the empty data storage section for erasing any data written thereto in response to a data write, and for returning data indicating an end of a file as empty data in response to a data read, as required by claim 1.

In other words, Olnowich merely teaches sending null data/words until a message is accepted/rejected, but fails to disclose or suggest returning data indicating an end of a file as empty data in response to a data read, as recited in claim 1.

Further, it is noted that amended independent claim 1 recites stream data processing apparatus including a control section for transmitting a change signal, wherein if a change signal is transmitted (i) to a transmitting-end processing section, then the transmitting-end processing section outputs a transmitting-end clear request, and (ii) to a receiving-end processing section, then the receiving-end processing section outputs a receiving-end clear request. Finally, claim 1 recites that a connection management section switches a write destination (between a data temporary storage section and an empty data storage section) and a read source (between the

data temporary storage section and the empty data storage section) based on whether the connection management section is in (ii) a receiving-end clear wait state which exists after the transmitting-end clear request is received and until the receiving-end clear request is received, or (ii) a transmitting-end clear wait state which exists after the receiving-end clear request is received and until the transmitting-end clear request is received. Olnowich and Kato, or any combination thereof, fails to disclose or suggest the above-discussed features of claim 1.

On the other hand, Kato teaches the use of a “ready 0” signal and a “ready 1” signal which control an operation of processors that transmit data between two systems (see col. 6, lines 51-60). More specifically, Kato teaches that the “ready 0” signal and the “ready 1” signal are signals indicating a “completion of a preparation of reception” and “a transmission of data for each data transfer unit,” respectively (see col. 6, lines 51-60; and Fig. 4). It is noted that the rejection equates the features of the above-mentioned change signal, as recited in claim 1, with the ready 0 and ready 1 signals as described by Kato.

However, it is apparent that Kato merely teaches the transmission of a signal indicating a completion of preparation for reception and the transmission of a signal indicating the transmission of data from a transfer unit, but fails to disclose or suggest switching a write destination (between a data temporary storage section and an empty data storage section) and a read source (between the data temporary storage section and the empty data storage section) based on whether the connection management section is in (i) a receiving-end clear wait state which exists after the transmitting-end clear request is received (as a result of a transmitted change signal) and until the receiving-end clear request is received (as a result of the transmitted change signal), or (ii) a transmitting-end clear wait state which exists after the receiving-end

clear request is received (as a result of the transmitted change signal) and until the transmitting-end clear request is received (as a result of the change signal), as required by claim 1.

Therefore, because of the above-mentioned distinctions it is believed clear that claim 1 and claims 2-11 that depend therefrom would not have been obvious or result from any combination of Olnowich and Kato.

Furthermore, there is no disclosure or suggestion in Olnowich and Kato or elsewhere in the prior art of record which would have caused a person of ordinary skill in the art to modify Olnowich and/or Kato to obtain the invention of independent claim 1. Accordingly, it is respectfully submitted that independent claim 1 and claims 2-11 that depend therefrom are clearly allowable over the prior art of record.

In view of the above amendments and remarks, it is submitted that the present application is now in condition for allowance and an early notification thereof is earnestly requested. The Examiner is invited to contact the undersigned by telephone to resolve any remaining issues.

Respectfully submitted,

Satoshi INAMI et al.

/Andrew L. Dunlap/

By: 2008.07.16 16:10:00 -04'00'

Andrew L. Dunlap
Registration No. 60,554
Attorney for Applicants

ALD/led
Washington, D.C. 20006-1021
Telephone (202) 721-8200
Facsimile (202) 721-8250
July 16, 2008